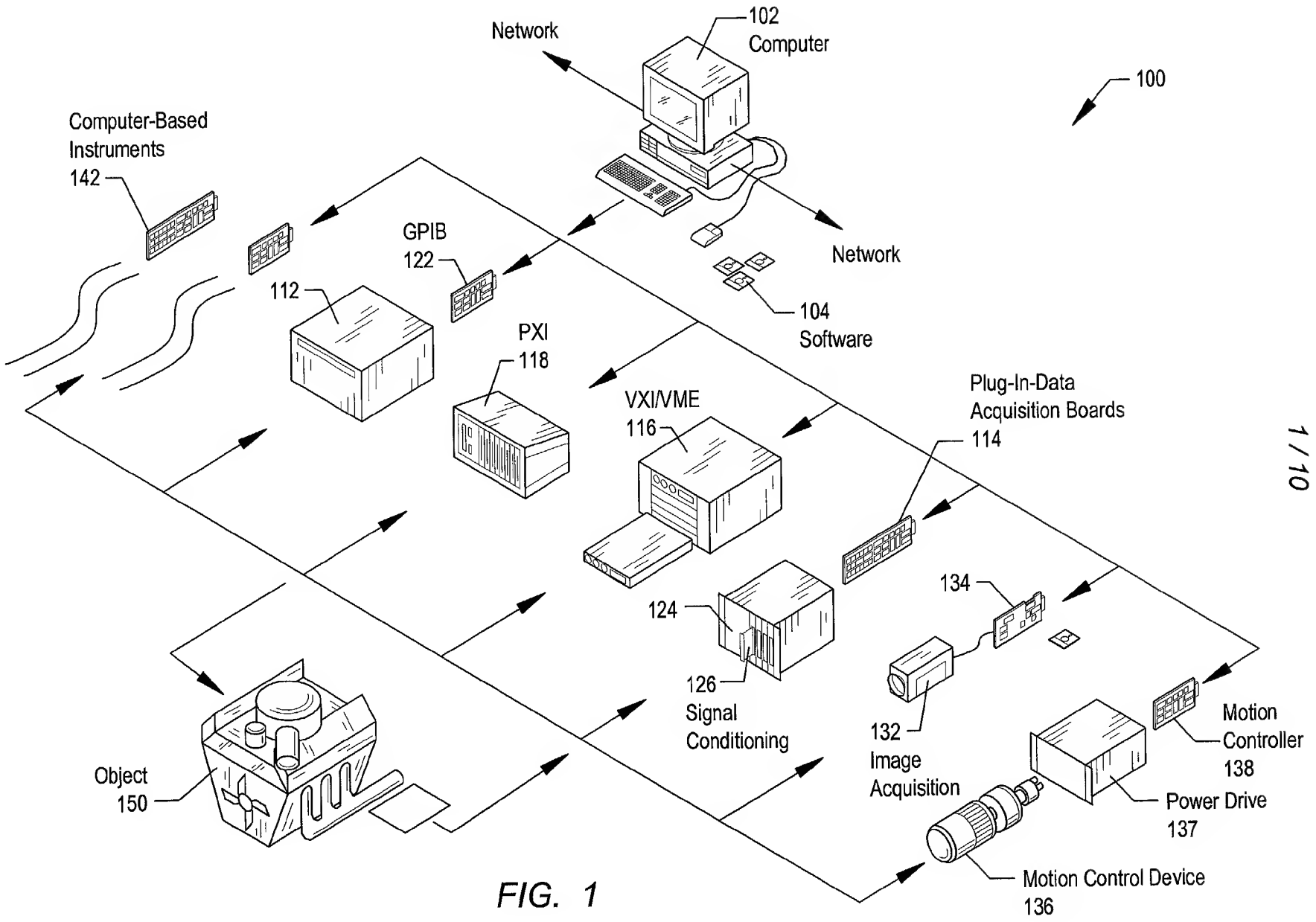
Network
102
Computer
100
Computer-Based
Instruments
142
GPIB
122
112
Network
104
Software
PXI
118
VXI/VME
116
Plug-In-Data
Acquisition Boards
114
124
134
126
Signal
Conditioning
132
Image
Acquisition
Motion
Controller
138
Power Drive
137
Object
150
Motion Control Device
136

FIG. 1

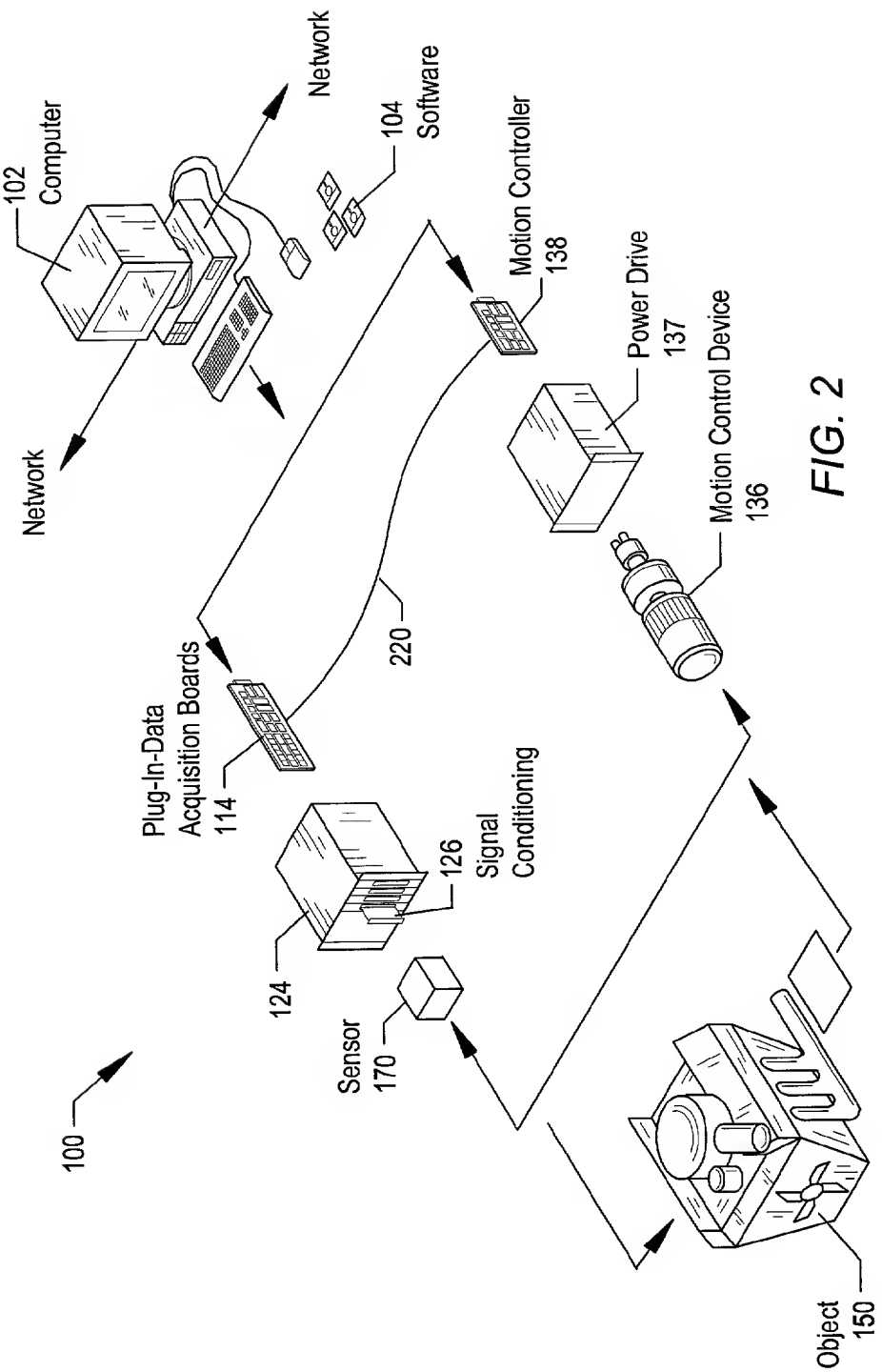
100
102
Computer
Network
Network
104
Software
Motion Controller
138
Power Drive
137
Motion Control Device
136
Plug-In-Data
Acquisition Boards
114
220
124
126
Signal
Conditioning
Sensor
170
Object
150

FIG. 2

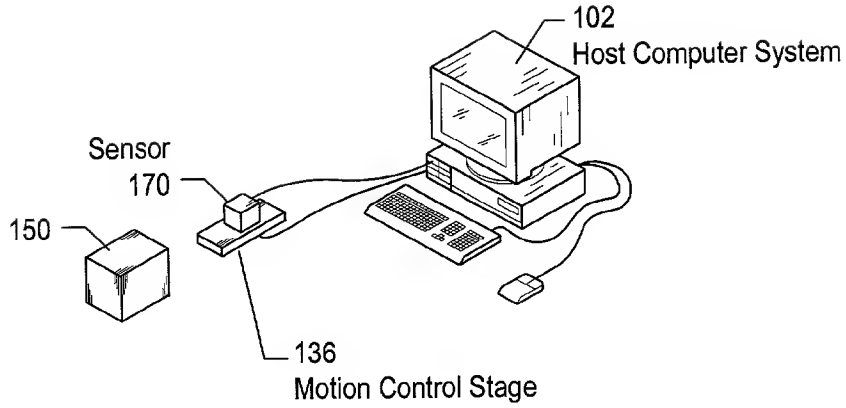
102
Host Computer System
Sensor
170
150
136
Motion Control Stage

FIG. 2A

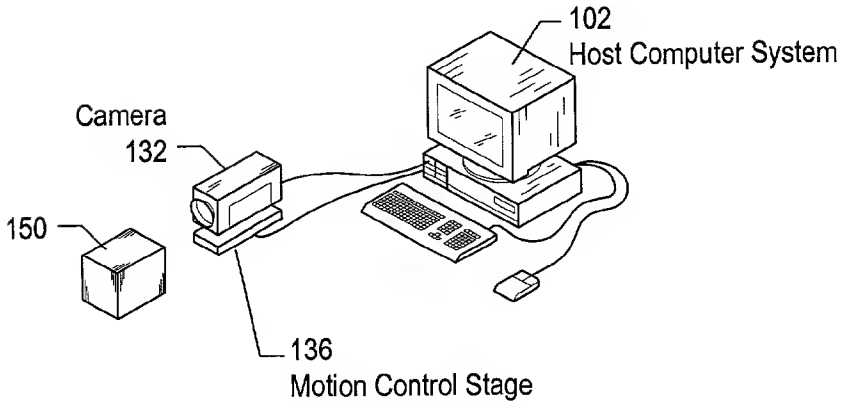
102
Host Computer System
Camera
132
150
136
Motion Control Stage

FIG. 2B

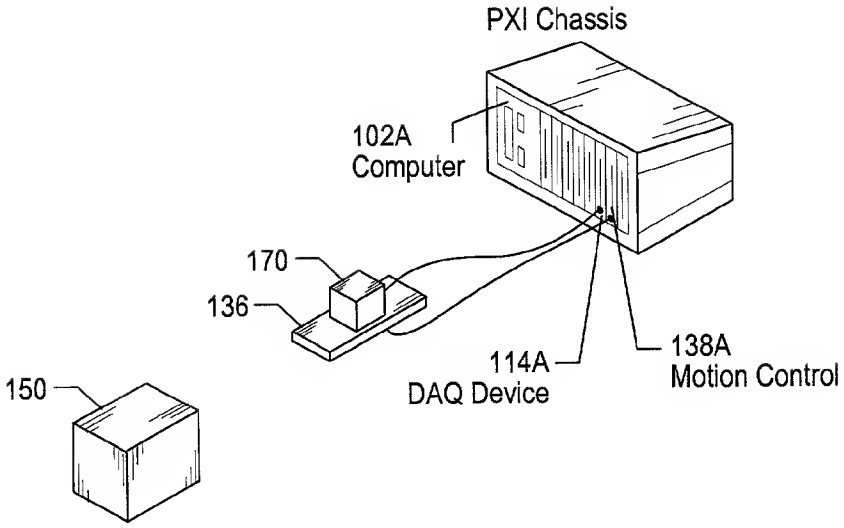
PXI Chassis
102A
Computer
170
136
150
114A
DAQ Device
138A
Motion Control

FIG. 3

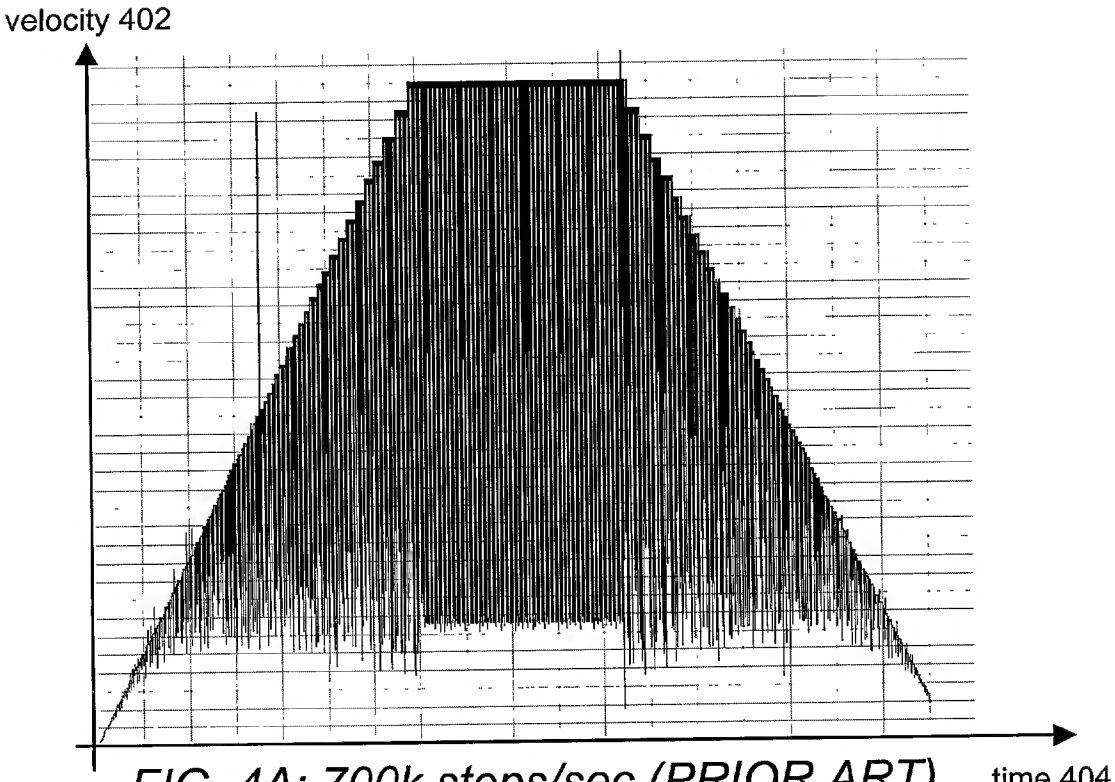

FIG. 4A: 700k steps/sec (PRIOR ART)

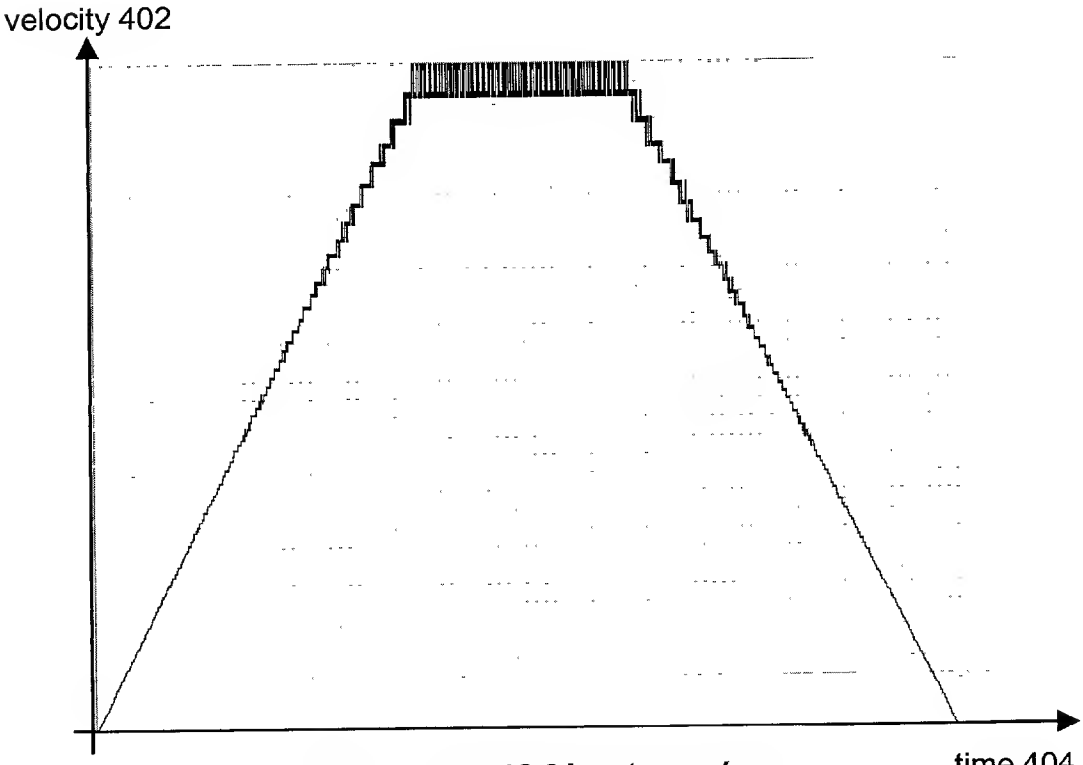

FIG. 4B: 700k steps/sec

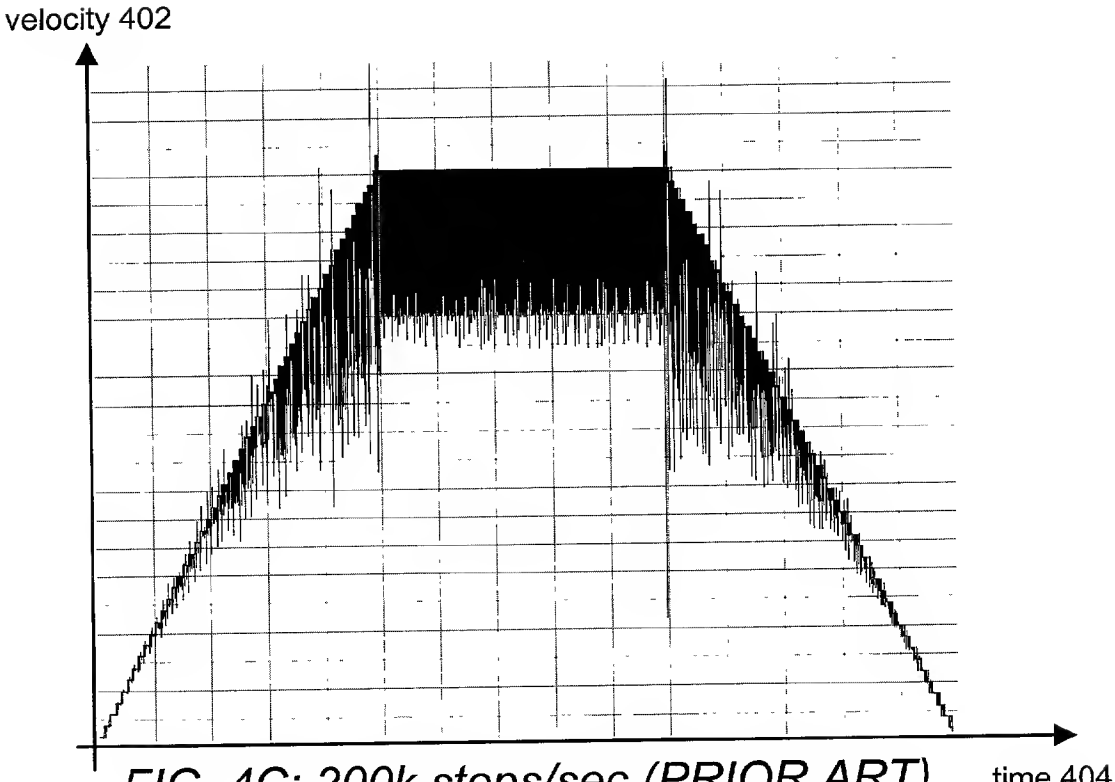

FIG. 4C: 200k steps/sec (PRIOR ART)

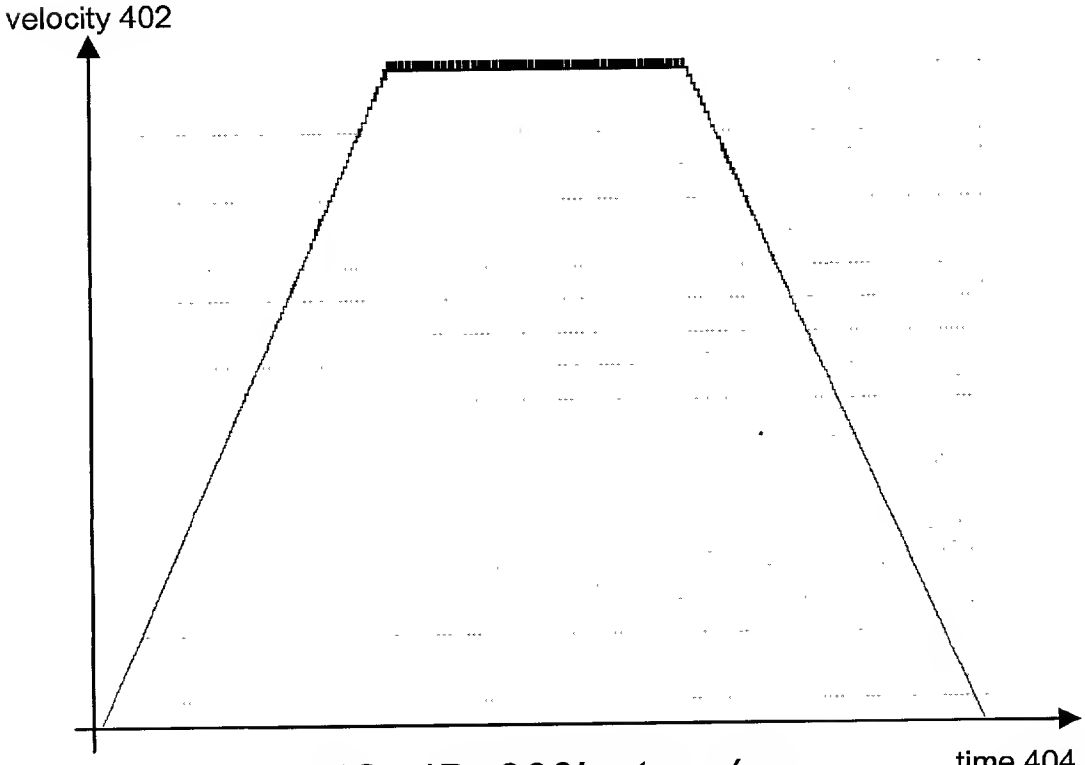

FIG. 4D: 200k steps/sec

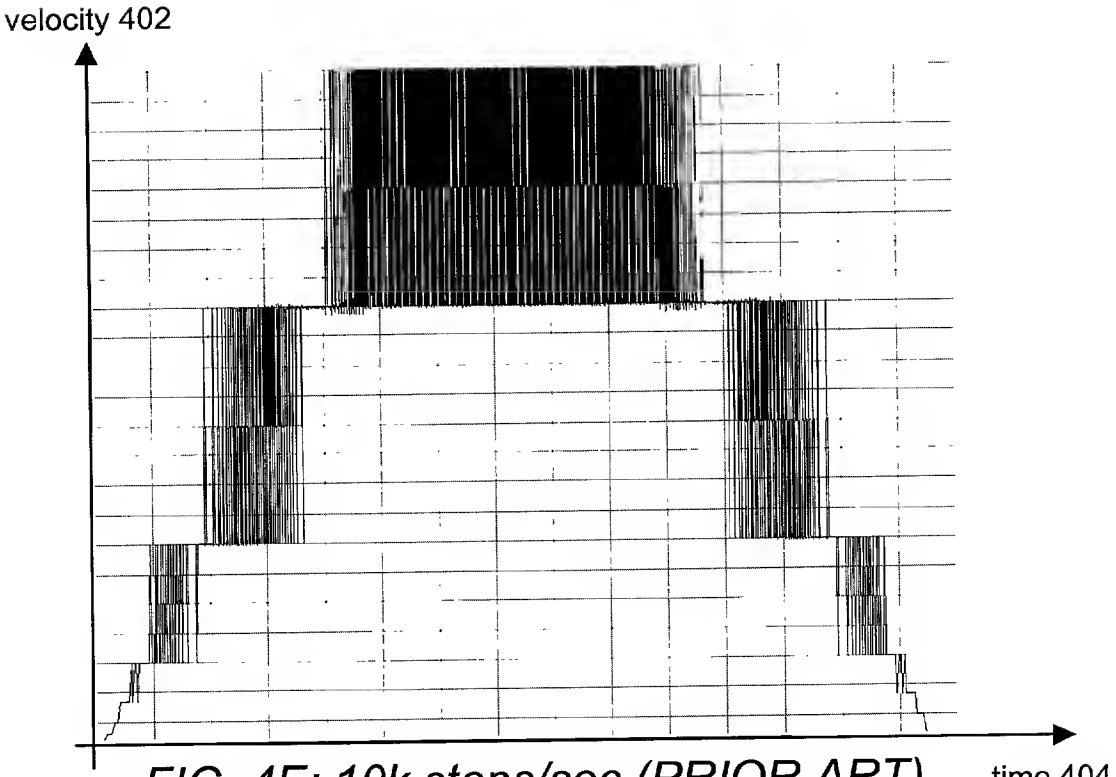

FIG. 4E: 10k steps/sec (PRIOR ART)

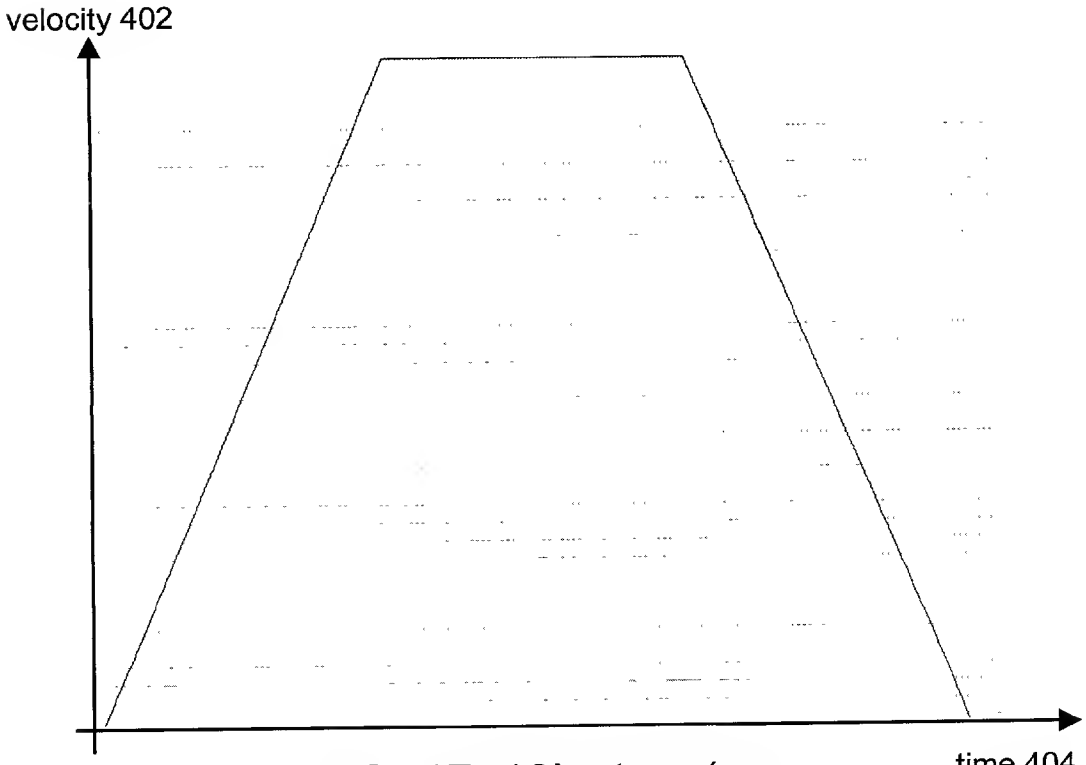

FIG. 4F: 10k steps/sec

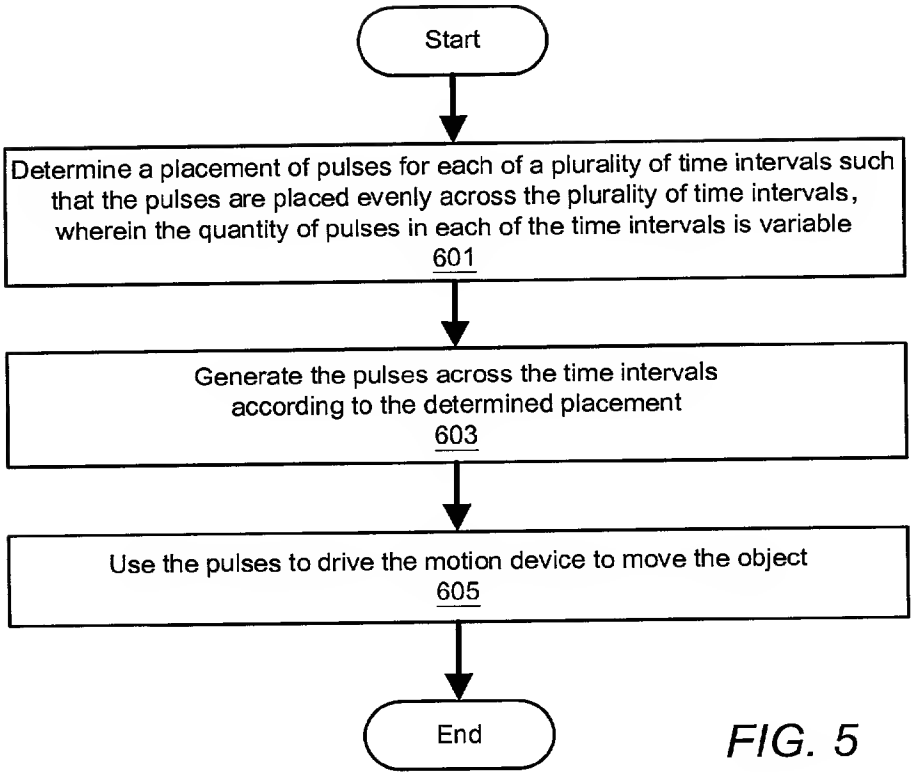
Start
Determine a placement of pulses for each of a plurality of time intervals such that the pulses are placed evenly across the plurality of time intervals, wherein the quantity of pulses in each of the time intervals is variable
601
Generate the pulses across the time intervals according to the determined placement
603
Use the pulses to drive the motion device to move the object
605
End

FIG. 5

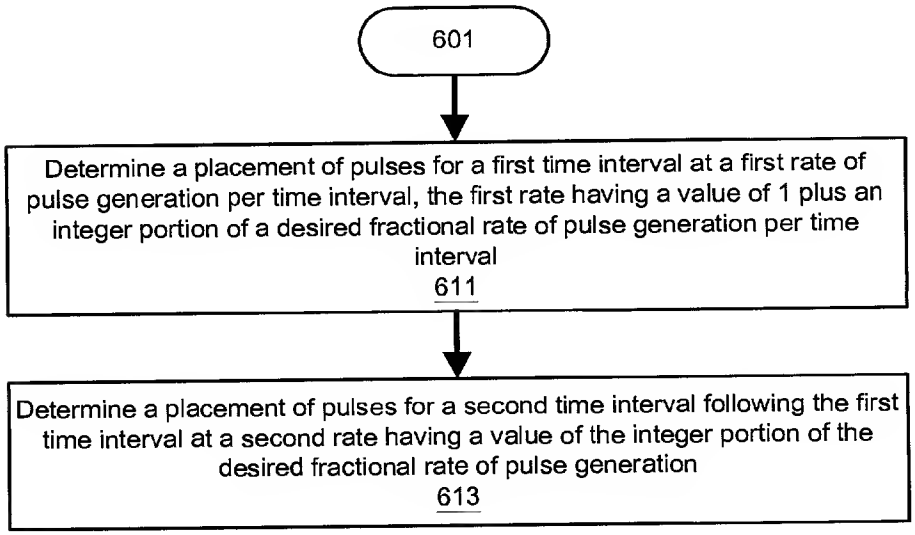
601
Determine a placement of pulses for a first time interval at a first rate of pulse generation per time interval, the first rate having a value of 1 plus an integer portion of a desired fractional rate of pulse generation per time interval
611
Determine a placement of pulses for a second time interval following the first time interval at a second rate having a value of the integer portion of the desired fractional rate of pulse generation
613

FIG. 6

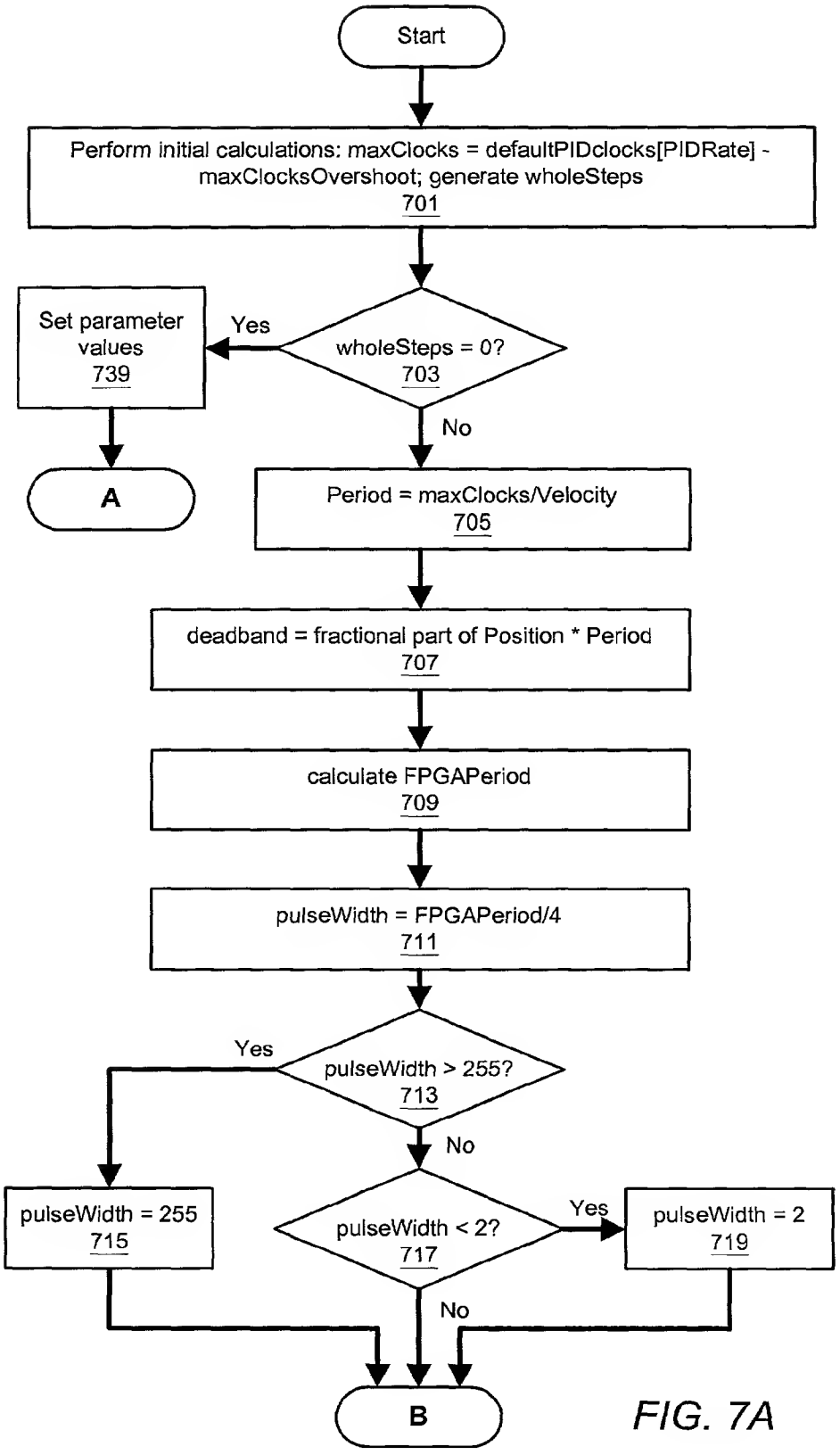
Start
Perform initial calculations: maxClocks = defaultPIDclocks[PIDRate] - maxClocksOvershoot; generate wholeSteps
701
wholeSteps = 0?
703
Yes
Set parameter values
739
A
No
Period = maxClocks/Velocity
705
deadband = fractional part of Position * Period
707
calculate FPGAPeriod
709
pulseWidth = FPGAPeriod/4
711
pulseWidth > 255?
713
Yes
No
pulseWidth = 255
715
pulseWidth < 2?
717
Yes
pulseWidth = 2
719
No
B

FIG. 7A

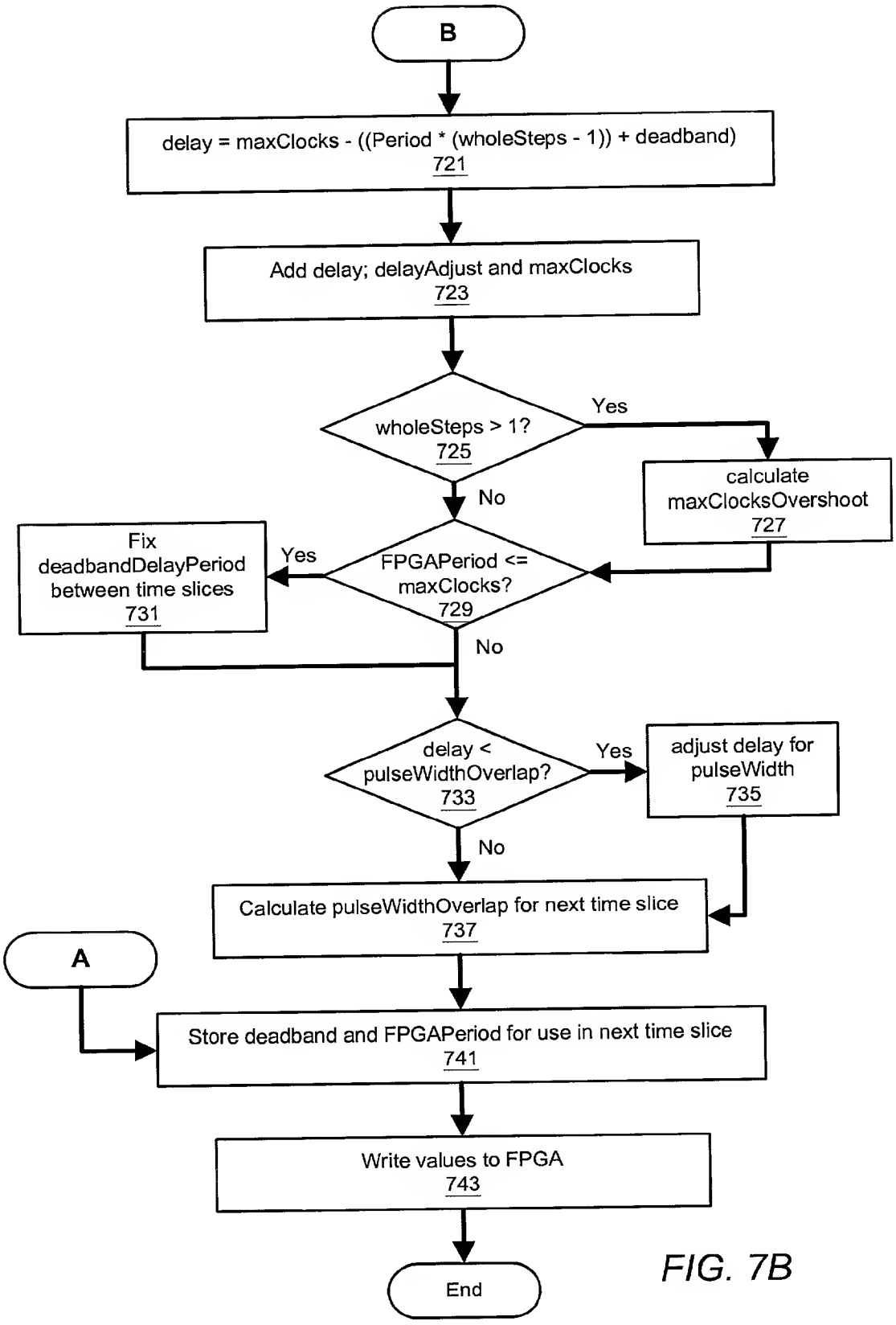
B
delay = maxClocks - ((Period * (wholeSteps - 1)) + deadband)
721
Add delay; delayAdjust and maxClocks
723
wholeSteps > 1?
725
Yes
No
calculate maxClocksOvershoot
727
Fix deadbandDelayPeriod between time slices
731
Yes
FPGAPeriod <= maxClocks?
729
No
delay < pulseWidthOverlap?
733
Yes
adjust delay for pulseWidth
735
No
Calculate pulseWidthOverlap for next time slice
737
A
Store deadband and FPGAPeriod for use in next time slice
741
Write values to FPGA
743
End

FIG. 7B